UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/686,537	10/16/2003	Hyun-kwon Chung	1101.0219	4036

89980 7590 08/03/2010
North Star Intellectual Property Law, PC
P.O. Box 34688
Washington, DC 20043

EXAMINER	
ZHEN, LI B	

ART UNIT	PAPER NUMBER
2194	

MAIL DATE	DELIVERY MODE
08/03/2010	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/686,537	CHUNG ET AL.	
	Examiner	Art Unit	
	LI B. ZHEN	2194	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *14 May 2010*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1,3,5-7,10,11,14,15,17-25 and 28-38* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1,3,5-7,10,11,14,15,17-25 and 28-38* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1, 3, 5-7, 10, 11, 14, 15, 17-25 and 28-38 are pending.

Response to Arguments

2. Applicant's arguments filed May 14, 2010 have been fully considered but they are not persuasive. In response to the previous office action, applicant argues:

(1) "...paragraphs [0066] and [0068] do not disclose or suggest that the preloader uses the 'report signal' to verify 'whether the markup document cannot be read due to an error, and whether the markup document is being read' as now recited in claim 1" (pages 13 – 14, 16, and 18 – 20);

(2) "it is submitted that paragraphs [0066] and [0068] do not disclose or suggest that the preloader performs the 'discard[ing]' and the 'indicat[ing]' using a 'report signal' generated by the file system I/O API, particularly since paragraph [0068] states that the preloader checks the file system I/O API after the preloader resumes preloading resources, which is after the preloader has already performed the 'discard[ing]' and the 'indicat[ing]'." (pages 14, 16, and 18 – 20);

(3) "the preloader 'discard[ing] the resource (or chunk of a resource) which it was currently trying to load' does not correspond to the preloader 'verify[ing] whether the markup document cannot be read due to an error' as recited in claim 1 as apparently alleged by the Office because the reason the preloader discards the resource (or chunk of a resource) which it was currently trying to load is because the preloader intentionally suspends itself from executing, rather than because the resource (or chunk of a

resource) 'cannot be read due to an error' as recited in claim 1" (pages 14, 16, and 18 – 20);

(4) "The Office considers paragraphs [0066]-[0068] of Jones to disclose 'staging the markup document for decoding in response to a retrieve signal' as recited in claim 32. However, the Office did not explain it considers paragraphs [0066]-[0068] of Jones to disclose this feature, and it is not seen where anything whatsoever in paragraphs [0066]-[0068] of Jones can reasonably be considered to disclose or suggest this feature." (page 17);

(5) "it is submitted that paragraph [0066] does not disclose or suggest that the preloaded does this 'in response to a discard signal'" (page 17)

As to arguments (1) and (2), examiner respectfully disagrees because Jones teaches generating a report signal used to identify a buffering state of the markup document using an application program interface (API) (I/O API; ¶66); using the report signal to verify whether the markup document has been successfully preloaded (checking a cache, memory, a file system I/O API, or some other location, to see if the file is already located local to the client; ¶68 and if the resource does exist locally, ¶69), whether the markup document cannot be read due to an error (I/O API indicates that a file is not located local to the client, ¶68 and 69; and preloader determines that the application has launched, the preloader may suspend itself from executing…preloader may discard the resource or chunk of a resource which it was currently trying to load, ¶66), and whether the markup document is being read (I/O API indicates that a file is

not located local to the client, ¶68 and 69; and indicate how much of the resource or chunk it was able to preload; ¶66); and outputting the information in response to the report signal (¶66, 68). The I/O API in Jones determines whether a file is already loaded. If the file is already loaded, the file has been successfully preloaded. If the file is not loaded, further information is used to determine the status of the file. When the I/O API indicates that a file is not located local to the client (¶68 and 69) and the preloader discards the resource or chunk of a resource which it was currently trying to load due to the launching of the application (¶66), the status of the file is that it cannot be read due to an error (the error being the interruption of the reading process due to the launching of the application). If the I/O API indicates that a file is not located local to the client (¶68 and 69) and the preloader indicate how much of the resource or chunk it was able to preload before suspending (¶66), then the status of the file is that the file is being read.

As to argument (3), examiner respectfully disagrees and notes that the claims only recite using the signal to verify whether "the markup document cannot be read due to an error" and does not identify the exact error. When the I/O API indicates that a file is not located local to the client (¶68 and 69) and the preloader discards the resource or chunk of a resource which it was currently trying to load due to the launching of the application (¶66), the status of the file is that it cannot be read due to an error (the error being the interruption of the reading process due to the launching of the application).

As to argument (4), it is noted that neither the claims nor the specification identify additional details of the 'staging' process. Examiner interprets "staging the markup document for decoding" as preloading the document for processing. Therefore, the preloading process disclosed in paragraphs [0066]-[0068] of Jones corresponds to the claimed staging process. In addition, Kanazawa teaches preloading markup documents for decoding (i.e. col. 15 lines 34 – 56; col. 17 line 64 – col. 18 line 23).

As to argument (5), Jones teaches that when the preloader determines that the application has launched, the preloader may suspend itself from executing and discard the resource (or chunk of a resource) which it was currently trying to load (¶66). The signal indicating that the application has launched corresponds to the claimed discard signal.

Claim Rejections - 35 USC § 103

3. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

4. **Claims 1, 3 and 5 – 7 are rejected under 35 U.S.C. 103(a) as being unpatentable over Kanazawa et al. (US 6,580,870 B1; "Kanazawa") in view of Jones et al. (US 2003/0220984 A1; "Jones") and further in view of Lamkin et al (US 7,448,021 B1; "Lamkin").**

5. As to claim 1, Kanazawa teaches a computer-readable storage medium usable with an apparatus comprising a buffer (abstract; col. 15 lines 46 – 57), the computer-readable storage medium having recorded thereon:

audio video (AV) data (abstract);

a markup document to be preloaded into the buffer of the apparatus to enable the apparatus to reproduce the AV data in an interactive mode selected by a user of the apparatus, wherein the markup document does not comprise the AV data or any other AV data (col. 15 lines 34 – 56; col. 17 lines 31 – 38; col. 20 lines 18 – 22); and

the apparatus to identify buffering state information of the markup document to be preloaded into the buffer of the apparatus, the buffering state information being used by the apparatus in reproducing the AV data in the interactive mode selected by the user (col. 15 lines 34 – 56; col. 17 line 64 – col. 18 line 23), wherein a report signal is used by the apparatus to verify whether the markup document has been success fully preloaded into the buffer (col. 18, lines 2 – 13). Although Kanazawa teaches the ability to identify the buffering state, it does not specifically teach that the identification is enabled by control information as claimed.

However, Jones teaches a buffer for preloading data (¶66, 72, 78, 88), identification is enabled by control information providing functionality (¶66, 68), the control information comprises an application program interface (API) (I/O API; ¶66) that generates a report signal used to identify a buffering state of the markup document (¶66, 68); and the report signal is used by the apparatus to verify whether the markup

document has been success fully preloaded into the buffer (checking a cache, memory, a file system I/O API, or some other location, to see if the file is already located local to the client; ¶68 and if the resource does exist locally, ¶69), whether the markup document cannot be read due to an error (I/O API indicates that a file is not located local to the client, ¶68 and 69; and preloader determines that the application has launched, the preloader may suspend itself from executing…preloader may discard the resource or chunk of a resource which it was currently trying to load, ¶66), and whether the markup document is being read (I/O API indicates that a file is not located local to the client, ¶68 and 69; and indicate how much of the resource or chunk it was able to preload; ¶66).

It would have been obvious to one of ordinary skill in the art at the time Applicant's invention was made to combine these teachings because Kanazawa teaches identifying the buffering state and Jones teaches a way to enable identification of the buffering state that can be used when implementing the disclosure of Kanazawa. Although Jones discloses that the control information may be stored on computer-readable medium such as a DVD, Jones does not specifically disclose storing the control information on a computer-readable medium that also includes AV data and a markup document.

However, Lamkin teaches a method for combining video/audio content with programmatic content (e.g. web pages) and software (col. 6, lines 4 – 61 and col. 8, lines 37 – 44). Lamkin also teaches that additional directories, runtime software, and programmatic content are added to the above directory structure, as needed, in order to

support additional hardware/software platforms, such as different types of personal computers and/or different operating systems, and consumer electronic devices, e.g., set top boxes and the like (col. 6, lines 55 – 62). As suggested by Lamkin, one of ordinary skill in the art would have been motivated to include control information as taught by Jones into the computer-readable medium (which includes AV data and markup documents) of Kanazawa. One of ordinary skill in the art would have been motivated to make the combination because this ensures that different types of personal computers and consumer electronic devices would have the necessary software to process the extra features (e.g. interlocking DVD video with HTML files) stored on the computer-readable medium.

6. As to claim 3, Kanazawa as modified (see rejections of claims 1 and 2) teaches the API comprises an [obj].isCached(URL, resType) API that generates the report signal, where the URL is a parameter indicating a file path of the markup document, and the resType is a parameter indicating an attribute of the markup document (Kanazawa: col. 15 lines 34 – 56; col. 17 line 64 – col. 18 line 23 and Jones: ¶66, 68). Kanazawa teaches determining whether a HTML file has been cached (steps S404 to S407; col. 17 line 64 – col. 18 line 23) but does not disclose the use of an API call. Jones discloses making an API call to determine whether a file is already located local to the client (paragraph 0068). The API call in Jones corresponds to the claimed isCached API and the URL and resType are interpreted as input parameters that are used to identify the markup document. The combination of Kanazawa and Jones would also include similar

parameters in order to uniquely identify each HTML file. For example, Kanazawa teaches identifying markup documents using URLs (i.e. col. 11, lines 48 – 62) and identifying markup document based on attribute information related to parental information (i.e. col. 5, lines 55 – 63). It would be obvious to a person of ordinary skill in the art that Kanazawa and Jones would use similar parameters as discloses in Kanazawa to identify and select certain markup documents.

7. As to claim 5, Kanazawa as modified teaches the control information further comprises an API that generates a fetch signal used to issue a command to preload the markup document (Kanazawa: col. 15 lines 34 – 56; col. 17 line 64 – col. 18 line 23).

8. As to claim 6, Kanazawa as modified teaches the API that generates the fetch signal returns a response indicating whether the command to preload the markup document has been successfully transmitted using the fetch signal (Kanazawa: col. 15 lines 34 – 56; col. 17 line 64 – col. 18 line 23).

9. As to claim 7, Kanazawa as modified teaches the control information further comprises an API that is used to determine whether preloading of the markup document is completed (Jones: ¶66, 68).

10. **Claims 10 and 11 are rejected under 35 U.S.C. 103(a) as being unpatentable over Kanazawa in view of Jones and Lamkin and further in view of US 20020088011 A1 ("Collart").**

11. As to claim 10, Kanazawa teaches wherein the interactive mode is a mode in which the AV data is interlocked with the markup document (col. 15 lines 32 – 45 and col. 11, lines 5 – 16); the apparatus is selectively operable in the interactive mode in which the AV data is interlocked markup document, and a non interactive video mode in which the AV data is displayed in the same manner as AV data recorded on a standard DVD (col. 6 lines 36 – 42; col. 15 lines 34 – 56); and the user of the apparatus selects between the interactive mode and the non interactive video mode (col. 15 lines 34 – 45). Kanazawa does not disclose the interactive mode is a mode in which the AV data is displayed in a display window defined by the markup document.

However, Collart teaches the interactive mode is a mode in which the AV data is displayed in a display window defined by the markup document (paragraphs 0117, 0121 – 0125) and the user of the apparatus selects between the interactive mode and the non interactive video mode (paragraph 0108).

It would have been obvious to a person of ordinary skill in the art at the time the invention was made to combine these teachings because this creates HTML-enhanced DVD-Video/Audio content that can play reliably across multiple playback platforms, ranging from computers to Internet-connected set-top devices (paragraph 0053 of Collart). This also allows content developers to create products that seamlessly

combine the Internet and/or other DVD-ROM capabilities with DVD-Video to create a richer, more interactive, and personalized entertainment experience for their customers (paragraph 0054 of Collart).

12. As to claim 11, Kanazawa as modified teaches a startup markup document (Collart: paragraph 0101) separate from the markup document to be preloaded into the buffer of the apparatus and comprising preloading instructions enabling the apparatus to preload the markup document (Collart: paragraphs 0105, 0217 and 0219) into the buffer of the apparatus (Kanazawa: col. 11 lines 5 – 11; col. 12 lines 43 – 48; col. 17 lines 31 – 38); wherein the selection of the interactive mode by the user causes the apparatus to read the startup markup document from the computer-readable storage medium and execute the preloading instructions to preload the markup document into the buffer of the apparatus (Kanazawa: col. 11 lines 5 – 11; col. 12 lines 43 – 48; col. 15 lines 34 – 56; col. 17 lines 31 – 38).

13. **Claims 14, 15, 17 – 21 and 25, 28 – 38 are rejected under 35 U.S.C. 103(a) as being unpatentable over Kanazawa; Koji et al. (US 6580870 B1; "Kanazawa") in view of Jones et al. (US 20030220984 A1; "Jones")**

14. As to claim 14, Kanazawa teaches an apparatus for reproducing audio video (AV) data using a markup document in an interactive mode selected by a user of the apparatus, comprising:

a buffer to buffer the markup document to enable the apparatus to reproduce the AV data in the interactive mode selected by the user (col. 15 lines 34 – 56; col. 17 lines 31 – 38); and

a buffer manager to manage the buffer to preload the markup document, the buffering state information being used by the apparatus in reproducing the AV data in the interactive mode selected by the user (col. 15 lines 34 – 56; col. 17 line 53 – col. 18 line 12).

Kanazawa fails to specifically teach output buffering state information of the buffer in response to a report signal; wherein the apparatus generates the report signal using an application program interface (API); and the report signal is used by the buffer manager to verify whether the markup document has been successfully preloaded into the buffer, whether the markup document cannot be read due to an error, and whether the markup document is being read.

However, Jones teaches output buffering state information of the buffer in response to a report signal (¶66, 68); wherein the apparatus generates the report signal using an application program interface (API) (I/O API; ¶66); and the report signal is used by the buffer manager to verify whether the markup document has been successfully preloaded into the buffer (checking a cache, memory, a file system I/O API, or some other location, to see if the file is already located local to the client; ¶68 and if the resource does exist locally, ¶69), whether the markup document cannot be read due to an error (I/O API indicates that a file is not located local to the client, ¶68 and 69; and preloader determines that the application has launched, the preloader may suspend

itself from executing…preloader may discard the resource or chunk of a resource which it was currently trying to load, ¶66), and whether the markup document is being read (I/O API indicates that a file is not located local to the client, ¶68 and 69; and indicate how much of the resource or chunk it was able to preload; ¶66).

It would have been obvious to one of ordinary skill in the art at the time Applicant's invention was made to combine these teachings because Kanazawa teaches identifying the buffering state and Jones teaches a way to enable identification of the buffering state that can be used when implementing the disclosure of Kanazawa.

15. As to claim 15, Kanazawa as modified teaches a content decoder to interpret the markup document, and generate the report signal using the API (Jones: ¶66, 68); wherein the buffer manager informs the content decoder of the buffering state information of the buffer in response to the report signal (Jones: ¶66, 68).

16. As to claim 17, Kanazawa as modified teaches a file path of the markup document as a parameter and an attribute of the markup document as parameters (Kanazawa: col. 11, lines 48 – 62, col. 5, lines 55 – 63 and; Jones: ¶66, 68).

17. As to claim 18, Kanazawa as modified teaches the API comprises an [obj].isCached(URL, resType) API that generates the report signal, where the URL is a parameter indicating a file path of the markup document, and the resType is a parameter indicating an attribute of the markup document (Kanazawa: col. 15 lines 34 –

56; col. 17 line 64 – col. 18 line 23 and Jones: ¶66, 68). Kanazawa teaches determining whether a HTML file has been cached (steps S404 to S407; col. 17 line 64 – col. 18 line 23) but does not disclose the use of an API call. Jones discloses making an API call to determine whether a file is already located local to the client (paragraph 0068). The API call in Jones corresponds to the claimed isCached API and the URL and resType are interpreted as input parameters that are used to identify the markup document. The combination of Kanazawa and Jones would also include similar parameters in order to uniquely identify each HTML file. For example, Kanazawa teaches identifying markup documents using URLs (i.e. col. 11, lines 48 – 62) and identifying markup document based on attribute information related to parental information (i.e. col. 5, lines 55 – 63). It would be obvious to a person of ordinary skill in the art that Kanazawa and Jones would use similar parameters as discloses in Kanazawa to identify and select certain markup documents.

18. As to claim 19, Kanazawa as modified teaches the buffer manager informs the content decoder of the buffering state of the markup document using the API (Jones: ¶66, 68).

19. As to claim 20, Kanazawa as modified teaches a content decoder to interpret the markup document (Kanazawa: col. 11, lines 15 – 39); wherein the buffer manager deletes the markup document from the buffer in response to a discard signal output from the content decoder (Jones: ¶49, 66).

20. As to claim 21, Kanazawa as modified teaches the content decoder generates the discard signal using a discard API (Jones: ¶49, 66 and 68).

21. As to claim 25, Kanazawa teaches a method of reproducing AV data in an interactive mode using a markup document, the method comprising:

buffering the markup document to preload the markup document (col. 15 lines 34 – 56; col. 17 lines 31 – 38; col. 20 lines 18 – 22). Kanazawa suggests indicating buffering state information (col. 15 lines 34 – 56; col. 17 line 64 – col. 18 line 23), but fails to specifically teach generating a report signal used to identify a buffering state of the markup document using an application program interface (API); using the report signal to verify whether the markup document has been successfully preloaded, whether the markup document cannot be read due to an error, and whether the markup document is being read; and outputting the information in response to the report signal.

However, Jones teaches generating a report signal used to identify a buffering state of the markup document using an application program interface (API) (I/O API; ¶66); using the report signal to verify whether the markup document has been successfully preloaded (checking a cache, memory, a file system I/O API, or some other location, to see if the file is already located local to the client; ¶68 and if the resource does exist locally, ¶69), whether the markup document cannot be read due to an error (I/O API indicates that a file is not located local to the client, ¶68 and 69; and preloader determines that the application has launched, the preloader may suspend itself from

executing…preloader may discard the resource or chunk of a resource which it was currently trying to load, ¶66), and whether the markup document is being read (I/O API indicates that a file is not located local to the client, ¶68 and 69; and indicate how much of the resource or chunk it was able to preload; ¶66); and outputting the information in response to the report signal (¶66, 68).

It would have been obvious to one of ordinary skill in the art at the time Applicant's invention was made to combine these teachings because Kanazawa teaches identifying the buffering state and Jones teaches a way to enable identification of the buffering state that can be used when implementing the disclosure of Kanazawa.

22. As to claim 28, Kanazawa as modified teaches the API comprise a file path and an attribute of the markup document as parameters (Kanazawa: col. 11, lines 48 – 62 and col. 5, lines 55 – 63; Jones: ¶66, 68).

23. As to claim 29, Kanazawa as modified teaches the API comprises an [obj].isCached(URL, resType) API that generates the report signal, where the URL is a parameter indicating a file path of the markup document, and the resType is a parameter indicating an attribute of the markup document (Kanazawa: col. 15 lines 34 – 56; col. 17 line 64 – col. 18 line 23 and Jones: ¶66, 68). Kanazawa teaches determining whether a HTML file has been cached (steps S404 to S407; col. 17 line 64 – col. 18 line 23) but does not disclose the use of an API call. Jones discloses making an API call to determine whether a file is already located local to the client (paragraph

0068). The API call in Jones corresponds to the claimed isCached API and the URL and resType are interpreted as input parameters that are used to identify the markup document. The combination of Kanazawa and Jones would also include similar parameters in order to uniquely identify each HTML file. For example, Kanazawa teaches identifying markup documents using URLs (i.e. col. 11, lines 48 – 62) and identifying markup document based on attribute information related to parental information (i.e. col. 5, lines 55 – 63). It would be obvious to a person of ordinary skill in the art that Kanazawa and Jones would use similar parameters as discloses in Kanazawa to identify and select certain markup documents.

24. As to claim 30, although the specific values of 0, 1 and 2 are not taught, Jones teaches the outputting of the buffering state information comprises returning a value in response to the markup document being successfully preloaded, returning a value in response to the markup document not being successfully preloaded, and returning a value in response to the markup document still being preloaded (¶ 66, 68).

25. As to claim 31, Kanazawa teaches reproducing the AV data in the interactive mode using the preloaded markup document (col. 15 lines 34 – 56).

26. As to claim 32, Kanazawa as modified teaches a method of managing a markup document for use in reproducing AV data in an interactive mode, the method comprising:

buffering the markup document to preload the markup document in response to a fetch signal (Kanazawa: col. 15 lines 34 – 56; col. 17 lines 31 – 38);

outputting a buffering state of the markup document in response to a report signal (Jones: ¶66 – 68);

staging the markup document for decoding in response to a retrieve signal (Jones: ¶66 – 68); and

deleting the markup document in response to a discard signal (Jones: ¶49, 66).

27. As to claim 33, Kanazawa teaches issuing a response indicating whether a command to preload the markup document included in the fetch signal has been successfully transmitted (col. 17, line 64 – col. 18, line 12).

28. As to claim 34, Kanazawa as modified teaches the outputting of the buffering state comprises returning a signal indicating whether preloading of the markup document has been completed (Jones: ¶66, 68).

29. As to claim 35, Kanazawa as modified teaches a method of managing a markup document for use in reproducing AV data in an interactive mode, the method comprising:

generating a fetch signal to preload the markup document (Kanazawa: col. 15 lines 34 – 56; col. 17 lines 31 – 38);

generating a report signal to determine a buffering state of the markup document (Jones: (¶66 – 68);

generating a retrieve signal to stage the markup document for decoding (Jones: (¶66 – 68); and

generating a discard signal to delete the markup document (Jones: ¶49, 66).

30. As to claim 36, Kanazawa as modified teaches generating a release signal in response to the markup document no longer being presented (Jones: ¶49, 66).

31. As to claim 37, Kanazawa as modified teaches wherein the outputting of a buffering state of the markup document in response to a report signal comprises: generating the report signal using an application program interface (API) (Jones: I/O API; ¶66); using the report signal to verify whether the markup document has been successfully preloaded (Jones: checking a cache, memory, a file system I/O API, or some other location, to see if the file is already located local to the client; ¶68), whether the markup document cannot be read due to an error (Jones: I/O API indicates that a file is not located local to the client, ¶68 and 69; and preloader determines that the application has launched, the preloader may suspend itself from executing…preloader may discard the resource or chunk of a resource which it was currently trying to load, ¶66), and whether the markup document is being read (Jones: I/O API indicates that a file is not located local to the client, ¶68 and 69; and indicate how much of the resource

or chunk it was able to preload; ¶66); and outputting the buffering state of the markup document in response to the report signal (Jones: ¶66, 68).

32. As to claim 38, Kanazawa as modified teaches wherein the generating of a report signal to determine a buffering state of the markup document comprises: generating the report signal using an application program interface (API) (Jones: I/O API; ¶66); using the report signal to verify whether the markup document has been successfully preloaded (Jones: checking a cache, memory, a file system I/O API, or some other location, to see if the file is already located local to the client; ¶68), whether the markup document cannot be read due to an error (Jones: I/O API indicates that a file is not located local to the client, ¶68 and 69; and preloader determines that the application has launched, the preloader may suspend itself from executing…preloader may discard the resource or chunk of a resource which it was currently trying to load, ¶66), and whether the markup document is being read (Jones: I/O API indicates that a file is not located local to the client, ¶68 and 69; and indicate how much of the resource or chunk it was able to preload; ¶66); and outputting the buffering state of the markup document in response to the report signal (Jones: ¶66, 68).

33. **Claims 22 and 23 are rejected under 35 U.S.C. 103(a) as being unpatentable over Kanazawa in view of Jones and further in view of US 20020088011 A1 ("Collart").**

34. As to claim 22, Kanazawa teaches the interactive mode is a mode in which the AV data is interlocked with the markup document (col. 15 lines 32 – 45 and col. 11, lines 5 – 16); the apparatus is selectively operable in the interactive mode in which the AV data is interlocked markup document, and a non interactive video mode in which the AV data is displayed in the same manner as AV data recorded on a standard DVD (col. 6 lines 36 – 42; col. 15 lines 34 – 56); and the user of the apparatus selects between the interactive mode and the non interactive video mode (col. 15 lines 34 – 45). Kanazawa does not disclose the interactive mode is a mode in which the AV data is displayed in a display window defined by the markup document.

However, Collart teaches the interactive mode is a mode in which the AV data is displayed in a display window defined by the markup document (paragraphs 0117, 0121 – 0125) and the user of the apparatus selects between the interactive mode and the non interactive video mode (paragraph 0108).

It would have been obvious to a person of ordinary skill in the art at the time the invention was made to combine these teachings because this creates HTML-enhanced DVD-Video/Audio content that can play reliably across multiple playback platforms, ranging from computers to Internet-connected set-top devices (paragraph 0053 of Collart). This also allows content developers to create products that seamlessly combine the Internet and/or other DVD-ROM capabilities with DVD-Video to create a richer, more interactive, and personalized entertainment experience for their customers (paragraph 0054 of Collart).

35. As to claim 23, Kanazawa as modified teaches an apparatus for recording and/or reproducing audio video (AV) data using a markup document in an interactive mode selected by a user of the apparatus before the apparatus reproduces any of the AV data (Collart: paragraph 0076), comprising:

an AV buffer to buffer the AV data (Kanazawa: col. 15 lines 34 – 56; col. 17 lines 31 – 38);

an AV reproduction engine to decode the AV data (Kanazawa: col. 15 lines 34 – 56; col. 17 lines 31 – 38);

an enhanced navigation (ENAV) buffer to preload the markup document before the apparatus reproduces any of the AV data to enable the apparatus to reproduce the AV data in the interactive mode selected by the user (Kanazawa: col. 15 lines 34 – 56; col. 17 lines 31 – 38);

an ENAV engine to interpret the markup document, and identify buffering state information of the markup document (Jones: ¶66, 68) in response to a report signal, the buffering state information being used by the apparatus in reproducing the AV data in the interactive mode selected by the user (Collart: paragraphs 0117, 0121 – 0125); and

an I/O manager to obtain the markup document (Kanazawa: col. 15 lines 34 – 56; col. 17 lines 31 – 38);

wherein:

the ENAV engine generates the report signal using an application program interface (API) (Jones: I/O API; ¶66 – ¶69): and

the report signal is used by the ENAV engine to verify whether the markup document has been successfully preloaded into the ENAV buffer (Jones: checking a cache, memory, a file system I/O API, or some other location, to see if the file is already located local to the client; ¶68), whether the markup document cannot be read due to an error (Jones: I/O API indicates that a file is not located local to the client, ¶68 and 69; and preloader determines that the application has launched, the preloader may suspend itself from executing…preloader may discard the resource or chunk of a resource which it was currently trying to load, ¶66), and whether the markup document is being read (Jones: I/O API indicates that a file is not located local to the client, ¶68 and 69; and indicate how much of the resource or chunk it was able to preload; ¶66).

36. **Claim 24 is rejected under 35 U.S.C. 103(a) as being unpatentable over Kanazawa, Jones and Collart and further in view of Silberschatz.**

37. As to claim 24, Kanazawa teaches obtaining the markup document, but fails to specifically teach blocked I/O and unblocked I/O. However, Silberschatz teaches the I/O manager uses a blocked I/O method to obtain data from a data storage medium (page 418 ¶5) and an unblocked I/O method to obtain data from a network (page 418 ¶2). It would have been obvious to one of ordinary skill in the art at the time Applicant's invention was made to combine these teachings because Kanazawa teaches what data needs to be transferred and Silberschatz teaches how to implement the data transfers.

Conclusion

38. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

CONTACT INFORMATION

39. Any inquiry concerning this communication or earlier communications from the examiner should be directed to LI B. ZHEN whose telephone number is (571)272-3768. The examiner can normally be reached on Mon - Fri, 8:30am - 5pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Hyung Sub Sough can be reached on 571-272-6799. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Li B. Zhen/
Primary Examiner, Art Unit 2194